Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 308 (2014) 87–107

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Strategies as Concurrent Processes

Simon Castellan[a] Jonathan Hayman[a] Marc Lasson[b]
Glynn Winskel[a]

[a] *Computer Laboratory, University of Cambridge, UK*

[b] *INRIA Paris-Rocquencourt, PiR2, Univ Paris Diderot, Sorbonne Paris Cité F-78153, Le Chesnay, France*

Abstract

Concurrent strategies are shown to support operations yielding an economic yet rich higher-order concurrent process language, which shares features both with process calculi and nondeterministic dataflow. Its operational semantics and 'may and must' equivalence require that we take internal (neutral) moves seriously, leading to the introduction of 'partial strategies' which may contain neutral moves. Through partial strategies, we can present a transition semantics for a language of strategies and can formulate their 'may and must' behaviour. While partial strategies compose, in a way extending that of strategies, in general composition introduces extra neutral moves; in particular, copy-cat is no longer strictly an identity w.r.t. composition. However, a simple extension of concurrent strategies (with stopping configurations) maintains the fact that they form a bicategory while still capturing 'may and must' behaviour.

Keywords: Concurrent game, event structures, bicategory

1 Introduction

There are several reasons for extending games and strategies, with behaviour based on trees, to concurrent games and strategies, based on event structures — the concurrent analogue of trees. One reason is to provide a foundation for a generalized domain theory, in which concurrent games and strategies take over the roles of domains and continuous functions. The motivation is to repair the divide between denotational and operational semantics and tackle anomalies like nondeterministic dataflow, which are beyond traditional domain theory. Another is that strategies are as potentially fundamental as relations and functions. It is surely because of our limited mental capacity, and not because of its unimportance, that the mathematical concept of strategy has been uncovered relatively late. It is hard to think about the successive contingencies involved in playing a game in the same way that is hard to think about interacting processes. Developing strategies in the extra generality demanded by concurrency reveals more clearly their essential nature and enables us to harness computer-science expertise in structure and concurrency in

http://dx.doi.org/10.1016/j.entcs.2014.10.006

their understanding and formalization.

The extra generality of concurrency reveals new structure and a mathematical robustness to the concept of strategy, in particular showing strategies are essentially special profunctors [19]. Profunctors themselves provide a rich framework in which to generalize domain theory, in a way that is arguably closer to that initiated by Dana Scott than game semantics [9,2]. However, the mathematical abstraction of profunctors comes at a price: it can be hard to give an operational reading to denotations as profunctors. There are examples of semantics of higher-order process languages and "strong correspondence" where elements of profunctor denotations correspond to derivations in an operational semantics [11,15]. But in general it is hard to extract operational semantics from the profunctor denotations alone because they have abstracted too far.

Connections between forms of strategy and process models have been the subject of a large body of prior research — see *e.g.* [8,6,7]. In this paper, we begin a study of concurrent strategies from the perspective of concurrent processes, considering how concurrent games and strategies are objects which we can program. They are shown to support operations yielding an economic yet rich higher-order concurrent process language, which shares features both with process calculi and nondeterministic dataflow. The operations allow recursion to be interpreted using a trace and novel duplication operation.

Indeed, seen from a concurrent-process perspective, in some respects concurrent strategies have abstracted too far. Both in obtaining a transition semantics for the language of strategies and in analysing its behaviour w.r.t. 'may and must' testing, we need to take internal (neutral) moves, introduced in the composition of strategies, seriously. Through the more refined model of *partial* concurrent strategies we obtain a correspondence between events of a strategy and derivations of atomic steps in a transition semantics. Via partial strategies we can justify a simple extension of concurrent strategies (with stopping configurations) which maintains the fact that they form a bicategory, while still capturing 'may and must' behaviour.

2 Event structures and their maps

An *event structure* comprises $(E, \mathrm{Con}, \leq)$, a set E of *events* which are partially ordered by $\leq$, the *causal dependency relation*, and a nonempty *consistency relation* Con consisting of finite subsets of E, which satisfy

$$\{e' \mid e' \leq e\} \text{ is finite for all } e \in E \qquad \{e\} \in \mathrm{Con} \text{ for all } e \in E$$

$$Y \subseteq X \in \mathrm{Con} \implies Y \in \mathrm{Con} \qquad X \in \mathrm{Con} \;\&\; e \leq e' \in X \implies X \cup \{e\} \in \mathrm{Con}.$$

The *configurations*, $\mathcal{C}^\infty(E)$, of an event structure E consist of those subsets $x \subseteq E$ which satisfy $\forall X \subseteq x.\ X \text{ is finite} \Rightarrow X \in \mathrm{Con}$ (*consistency*) and $\forall e, e'.\ e' \leq e \in x \implies e' \in x.$ (*down-closure*). Often we shall be concerned with just the finite configurations of an event structure. We write $\mathcal{C}(E)$ for the *finite* configurations of an event structure E.

We say an event structure is *elementary* when the consistency relation consists of all finite subsets of events. Two events which are both consistent and incomparable w.r.t. causal dependency in an event structure are regarded as *concurrent*. In games the relation of *immediate* dependency $e \rightarrowtriangle e'$, meaning e and e' are distinct with $e \leq e'$ and no event in between, will play a very important role. For $X \subseteq E$ we write $[X]$ for $\{e \in E \mid \exists e' \in X.\ e \leq e'\}$, the down-closure of X; note if $X \in \mathrm{Con}$, then $[X] \in \mathrm{Con}$. We write $[a]$ for $[\{a\}]$ where $a \in E$. For configurations x, y, we use $x \mathrel{-\!\!\subset} y$ to mean y covers x, *i.e.* $x \subset y$ with nothing in between, and $x \overset{e}{\mathrel{-\!\!\subset}} y$ to mean $x \cup \{e\} = y$ for an event $e \notin x$. We sometimes use $x \overset{e}{\mathrel{-\!\!\subset}}$, expressing that event e is enabled at configuration x, when $x \overset{e}{\mathrel{-\!\!\subset}} y$ for some configuration y.

Certain 'structural' maps of event structures, which have a long history [14], play a key role in the development of nondeterministic concurrent strategies. A *map* of event structures $f : E \to E'$ is a partial function $f : E \rightharpoonup E'$ such that $fx \in \mathcal{C}^\infty(E')$ for all $x \in \mathcal{C}^\infty(E)$, and for all $e_1, e_2 \in x$

$$f(e_1) = f(e_2)\ \&\ f(e_1), f(e_1) \text{ both defined} \implies e_1 = e_2\,.$$

Above, it is sufficient to restrict to finite configurations. Note that, when f is total, it restricts to a bijection $x \cong fx$ for any $x \in \mathcal{C}^\infty(E)$. A total map is *rigid* when it preserves causal dependency.

A map $f : E \to E'$ of event structures has *partial-total factorization* as a composition $E \overset{p}{\longrightarrow} E{\downarrow}V \overset{t}{\longrightarrow} E'$ where $V =_{\mathrm{def}} \{e \in E \mid f(e) \text{ is defined}\}$ is the domain of definition of f; the event structure $E{\downarrow}V =_{\mathrm{def}} (V, \leq_V, \mathrm{Con}_V)$, where $v \leq_V v'$ iff $v \leq v'$ & $v, v' \in V$ and $X \in \mathrm{Con}_V$ iff $X \in \mathrm{Con}$ & $X \subseteq V$; the *partial* map $p : E \to E{\downarrow}V$ acts as identity on V and is undefined otherwise; and the *total* map $t : E{\downarrow}V \to E'$, called the *defined part* of f, acts as f. The event structure $E{\downarrow}V$ is the *projection* of E to V.

It shall be convenient to construct event structures using *rigid families*.

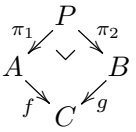
Proposition 2.1 *Let $\mathcal{Q}$ be a non-empty family of finite partial orders closed under rigid inclusions, i.e. if $q \in \mathcal{Q}$ and $q' \hookrightarrow q$ is a rigid inclusion (regarded as a map of elementary event structures) then $q' \in \mathcal{Q}$. The family $\mathcal{Q}$ determines an event structure $\Pr(\mathcal{Q}) =_{\mathrm{def}} (P, \leq, \mathrm{Con})$ as follows:*

- *the events P are primes, i.e. finite partial orders in $\mathcal{Q}$ with a top element;*
- *the causal dependency relation $p' \leq p$ holds precisely when there is a rigid inclusion from $p' \hookrightarrow p$;*
- *a finite subset $X \subseteq P$ is consistent, $X \in \mathrm{Con}$, iff there is $q \in \mathcal{Q}$ and rigid inclusions $p \hookrightarrow q$ for all $p \in X$.*

If $x \in \mathcal{C}(P)$ then $\bigcup x$, the union of the partial orders in x, is in $\mathcal{Q}$. The function $x \mapsto \bigcup x$ is an order-isomorphism from $\mathcal{C}(P)$, ordered by inclusion, to $\mathcal{Q}$, ordered by rigid inclusions.

Pullbacks of total maps Maps $f : A \to C$ and $g : B \to C$ have pullbacks in the category of event structures, and are simple to describe in the case where f and

g are total. We give a pullback object P along with projections π_1 and π_2 as shown.



In this situation, finite configurations of P correspond to the composite bijections

$$\theta : x \cong fx = gy \cong y$$

between configurations $x \in \mathcal{C}(A)$ and $y \in \mathcal{C}(B)$ s.t. $fx = gy$ for which the transitive relation generated on θ by $(a, b) \leq (a', b')$ if $a \leq_A a'$ or $b \leq_B b'$ is a partial order; the correspondence taking $z \in \mathcal{C}(P)$ to the composite bijection $\pi_1 z \cong f\pi_1 z = g\pi_2 z \cong \pi_2 z$ respects inclusion.

2.1 Affine maps

In considering the dynamics of processes we shall need to relate a process to a process it may become. For this we generalize the earlier structural maps of event structures to *affine* maps, in which we need no longer preserve the empty configuration [16].

Definition 2.2 Let A be an event structure. Let $x \in \mathcal{C}^\infty(A)$. Write A/x for the event structure which remains after the occurrence of x. Precisely, A/x comprises

- events, $\{a \in A \setminus x \mid x \cup [a]_A \in \mathcal{C}^\infty(A)\}$,
- consistency relation, $X \in \mathrm{Con}$ iff $X \subseteq_{\mathrm{fin}} A/x$ & $x \cup [X]_A \in \mathcal{C}^\infty(A)$, and
- causal dependency, the restriction of that on A.

We extend the notation to configurations regarding them as elementary event structures. If $y \in \mathcal{C}^\infty(A)$ with $x \subseteq y$ then by y/x we mean the configuration $y \setminus x \in \mathcal{C}^\infty(A/x)$. In the case of a singleton configuration $\{a\}$ of A — when a is an *initial* event of A — we shall often write A/a and x/a instead of $A/\{a\}$ and $x/\{a\}$.

An *affine* map of event structures f from A to B comprises a pair (f_0, f_1) where $f_0 \in \mathcal{C}(B)$ and f_1 is a map of event structures $f_1 : A \to B/f_0$. It determines a function from $\mathcal{C}(A)$ to $\mathcal{C}(B)$ given by $fx = f_0 \cup f_1 x$ for $x \in \mathcal{C}(A)$. The allied f_0 and f_1 can be recovered from the action of f on configurations: $f_0 = f\varnothing$ and f_1 is that unique map of event structures $f_1 : A \to B/f\varnothing$ which on configurations $x \in \mathcal{C}(A)$ returns $fx/f\varnothing$. It is simplest to describe the composition gf of affine maps $f = (f_0, f_1)$ from A to B and $g = (g_0, g_1)$ from B to C in terms of its action on configurations: the composition takes a configuration $x \in \mathcal{C}(A)$ to $g(f\,x)$. Alternatively, the composition gf can be described as comprising $(g_0 \cup g_1 f_0, h)$ where h is that unique map of event structures $h : A \to C/(g_0 \cup g_1 f_0)$ which sends $x \in \mathcal{C}(A)$ to $g_1(f_0 \cup f_1 x)/g_1 f_0$. Note that traditional maps can be identified with those special affine maps (f_0, f_1) in which $f_0 = \varnothing$. We reserve the term 'map' for the traditional structural maps of event structure and shall say explicitly when a map is affine.

3 Concurrent games and strategies

A *game* is represented by an event structure A in which an event $a \in A$ carries a polarity $pol(a)$, $+$ for Player and $-$ for Opponent. Maps and affine maps of event structures extend to event structures with polarity: their underlying partial functions on events are required, where defined, to preserve polarity. A number of constructions on games will play an important role in the coming semantics:
Dual $A^{\perp}$, of an event structure with polarity A is a copy of the event structure A with a reversal of polarities.
Simple parallel composition $A \| B$, by juxtaposition. Its unit is the empty game $\varnothing$. More generally, we can define, $\|_{i \in I} A_i$, the simple parallel composition of a indexed family of games, in which the set of events comprises the disjoint union $\bigcup_{1 \leq i \leq m} \{i\} \times A_i$.
Sums and recursive definitions Sums $\Sigma_{i \in I} A_i$ of an indexed family of games, which coincide with coproducts in the categories of event structures, are obtained in a similar way to simple parallel compositions, but now with events from distinct components being inconsistent (*i.e.* no set in the consistency relation contains elements from distinct components). We shall not make explicit use of recursively-defined games, but they are dealt with in exactly the same way as recursively-defined event structures [14].

3.1 A bicategory of games and strategies

A (nondeterministic concurrent) *strategy in* a game A is represented by a total map of event structures $\sigma : S \to A$ which preserves polarities and is
Receptive: if $\sigma x \overset{a}{\longrightarrow}$ & $pol_A(a) = -$ then there is unique $s \in S$ s.t. $x \overset{s}{\longrightarrow}$ & $\sigma(s) = a$;
Innocent: if $s \rightarrowtriangle_S s'$ & $(pol(s) = + \text{ or } pol(s') = -)$ then $\sigma(s) \rightarrowtriangle_A \sigma(s')$.

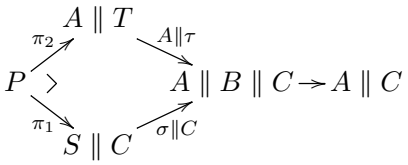
Receptivity expresses that Player cannot hinder moves of Opponent, while innocence says a strategy should only adjoin immediate causal dependencies of the form $\ominus \rightarrowtriangle \oplus$. A map between strategies from $\sigma : S \to A$ to $\sigma' : S' \to A$ is a total map $f : S \to S'$ of event structures with polarity such that $\sigma = \sigma' f$. Accordingly, the strategies are isomorphic iff f is an isomorphism of event structures.

The conditions of receptivity and innocence are necessary and sufficient to ensure that the copy-cat strategy behaves as identity w.r.t. composition [12], which we now proceed to define.

We follow Conway and Joyal, and define a *strategy from* a game A *to* a game B, written $\sigma : A \nrightarrow B$, as a strategy σ in the game $A^{\perp} \| B$.

Let $\sigma : S \to A^{\perp} \| B$, $\tau : T \to B^{\perp} \| C$ be strategies. Their composition is defined via the pullback drawn below. Ignoring polarities, the composite partial map has defined part $T \odot S$, which yields the composition of strategies $\tau \odot \sigma : T \odot S \to A^{\perp} \| C$

once polarities are reinstated.



Let A be a game. The copy-cat strategy from A to A is a total map $\gamma_A : !!C_A \to A^\perp \| A$, based on the idea that Player moves, of +ve polarity, always copy previous corresponding moves of Opponent, of −ve polarity. For $c \in A^\perp \| A$ we use $\overline{c}$ to mean the corresponding copy of c, of opposite polarity, in the alternative component. Define $!!C_A$ to comprise the event structure with polarity $A^\perp \| A$ together with extra causal dependencies $\overline{c} \leq_{!!C_A} c$ for all events c with $pol_{A^\perp \| A}(c) = +$. A finite subset of $!!C_A$ is consistent if its down-closure is consistent in $A^\perp \| A$.

The characterisation of configurations of $!!C_A$ reveals an important partial order on configurations of A. Let x and y be configurations of an event structure with polarity. Write $x \subseteq^- y$ to mean $x \subseteq y$ and $pol(y \setminus x) \subseteq \{-\}$, *i.e.* the configuration y extends the configuration x solely by events of −ve polarity. Similarly, $x \subseteq^+ y$ means $x \subseteq y$ and $pol(y \setminus x) \subseteq \{+\}$. Use $\supseteq^-$ to denote the converse order to $\subseteq^-$. Define the *Scott order* on configurations by $x \sqsubseteq y$ iff $x \supseteq^- \cdot \subseteq^+ \cdot \supseteq^- \cdots \supseteq^- \cdot \subseteq^+ y$. Then, $\sqsubseteq$ is a partial order and part of a factorization system: if $x \sqsubseteq y$ then $\exists! z.\ x \supseteq^- z \subseteq^+ y$.

Proposition 3.1 *[19] Let A be a game. Then, $x \in \mathcal{C}(!!C_A)$ iff $x_2 \sqsubseteq_A x_1$, where $x_1 \in \mathcal{C}(A^\perp)$ and $x_2 \in \mathcal{C}(A)$ are the projections of $x \in \mathcal{C}(A^\perp \| A)$ to its components.*

Theorem 3.2 *[19] Strategies $\sigma : S \to A$ correspond to discrete fibrations denoted $\sigma\text{“}: (\mathcal{C}(S), \sqsubseteq_S) \to (\mathcal{C}(A), \sqsubseteq_A)$, preserving $\supseteq^-$, $\subseteq^+$ and $\varnothing$.*

The theorem says we can view strategies in a game as (certain) discrete fibrations, so equivalently as presheaves over finite configurations with the Scott order. In particular, a strategy from a game A to a game B corresponds to a presheaf over $(\mathcal{C}(A^\perp \| B), \sqsubseteq_{A^\perp \| B}) \cong (\mathcal{C}(A), \sqsubseteq_A)^{\mathrm{op}} \times (\mathcal{C}(B), \sqsubseteq_B)$, so to a profunctor from $(\mathcal{C}(A), \sqsubseteq_A)$ to $(\mathcal{C}(B), \sqsubseteq_B)$. This correspondence yields a lax functor from strategies to profunctors. The view of strategies as (special) profunctors — explained further in [19] — will guide our later work.

We obtain a bicategory of concurrent games and strategies in which the objects are event structures with polarity — the games, the arrows from A to B are strategies $\sigma : A \nrightarrow B$ and the 2-cells are maps of strategies. The vertical composition of 2-cells is the usual composition of maps. Horizontal composition is given by the composition of strategies $\odot$ (which extends to a functor on 2-cells via the universality of pullback).

A strategy in the game $A^\perp \| B$ corresponds to a strategy in the game $(B^\perp)^\perp \| A^\perp$. Hence a strategy $A \nrightarrow B$ corresponds to a dual strategy $B^\perp \nrightarrow A^\perp$. The bicategory is rich in structure, in particular, it is compact-closed (so has a trace, a feedback operation).

3.2 Operations on strategies and duplication

Beyond composition there are many other useful operations on strategies. Several of these have appeared previously in, for example, establishing determinacy [3] or the value theorem for games with pay-off [4] where proofs often hinge on constructing appropriate strategies.

We can form the *sum of strategies* $[]_{i\in I}\,\sigma_i$ of a family of strategies $\sigma_i : S_i \to A$, $i \in I$, in a common game A [4]. This is formed as the sum of the event structures S_i but where the initial $-$ve events are identified to maintain receptivity. A sum of strategies only commits to a particular component strategy once a Player move is made there. The empty sum $\bot$ essentially comprises the initial segment of the game A consisting of all the initial $-$ve events of A.

The pullback of a strategy σ across a (possibly partial) map f of event structures is itself a strategy $f^*\sigma$ [18]:

$$\begin{array}{ccc} S' & \longrightarrow & S \\ {\scriptstyle f^*\sigma}\downarrow & \lrcorner & \downarrow{\scriptstyle \sigma} \\ A & \xrightarrow[f]{} & B\,. \end{array}$$

This operation can adjoin extra events and causal links to the original strategy; it subsumes, for example, operations on strategies in which we prefix an initial event.

We shall also use a previously unnoticed strategy δ_A that exists from a game A to $A\|A$, and expresses a form of duplication. Like copy-cat it introduces extra 'causal wiring' but its definition is much more subtle, though is easy to describe in special cases. For example if the game A comprises a single Player move $\oplus$ the strategy $\delta_A : A \mapsto\!\!\!\!\rightarrow A\|A$ takes the form $\begin{array}{ccc} & \nearrow & \oplus \\ \ominus & & \\ & \searrow & \oplus \end{array}$. A similar construction applies in the case where all the moves of a game are that of Player. If the game A comprises a single Opponent move $\ominus$ then $\delta_A : A \mapsto\!\!\!\!\rightarrow A\|A$ takes the form $\begin{array}{ccc} \oplus & \leftarrow & \ominus \\ & \wr & \\ \oplus & \leftarrow & \ominus \end{array}$, where the wiggly line indicates the inconsistency between the two Player moves.

Duplication The definition of $\delta_A : A \mapsto\!\!\!\!\rightarrow A\|A$ in general is via rigid families. For each triple (x, y_1, y_2), where $x \in \mathcal{C}(A^\perp)$ and $y_1, y_2 \in \mathcal{C}(A)$, which is *balanced*, *i.e.*

$$\forall a \in y_1 \cup y_2.\ pol_A(a) = + \implies a \in x \text{ and } \forall a \in x.\ pol_{A^\perp}(a) = + \implies a \in y_1 \text{ or } a \in y_2,$$

and *choice* function $\chi : x^+ \to \{1, 2\}$, from the positive events of x denoted by x^+, such that $\chi(a) = 1 \implies a \in y_1$ and $\chi(a) = 2 \implies a \in y_2$, the order $q(x, y_1, y_2; \chi)$ is defined to have underlying set $\{0\} \times x \cup \{1\} \times y_1 \cup \{2\} \times y_2$ with order generated by that inherited from $A^\perp\|A\|A$ together with

$$\begin{array}{l} \{((0,a),(1,a)) \mid a \in y_1\} \quad \cup \quad \{((0,a),(2,a)) \mid a \in y_2\} \quad \cup \\ \{((\chi(a),a),(0,a)) \mid a \in x \ \&\ pol_{A^\perp}(a) = +\}\,. \end{array}$$

The rigid family $\mathcal{Q}$ consists of all such $q(x, y_1, y_2; \chi)$ for balanced (x, y_1, y_2) and choice functions χ. From $\mathcal{Q}$ we obtain the event structure $\Pr(\mathcal{Q})$ in which events are prime orders, *i.e.* with a top element; events of $\Pr(\mathcal{Q})$ inherit the polarity of their top elements to obtain an event structure with polarity. We define the strategy $\delta_A : A \mapsto\!\!\!\!\rightarrow A\|A$ to be the map $\Pr(\mathcal{Q}) \to A^\perp\|A\|A$ sending a prime to its top

element. We remark that the meaning of a triple of configurations x, y_1, y_2 of C being balanced is almost $y_1 \cup y_2 \sqsubseteq_C x$ but is not this in general as $y_1 \cup y_2$ need not itself be a configuration of C.

The operation δ_A forms a comonoid with counit $\bot : A \mapsto \varnothing$.

4 A language for strategies

We describe, somewhat schematically, a language for describing strategies based on the constructions above. In fact, it is based on an earlier language for profunctors [15], taking advantage of the view of strategies as special profunctors.

4.1 Types

Types are games $A, B, C, \cdots$. We have type operations corresponding to the operations on games of forming the dual $A^\bot$, simple parallel composition $A \| B$, sum $\Sigma_{i \in I} A_i$ as well as recursively-defined types $\mu X.A(X)$, although we shall largely ignore the latter as it rests on well-established techniques [14].

One way to relate types is through the affine maps between them. There will be operations for shifting between types related by affine maps (described by configuration expressions), enabling us to shift strategies either forwards or backwards across affine maps.

A *type environment* is a finite partial function from variables to types, for convenience written typically as $\Gamma \equiv x_1 : A_1, \cdots, x_m : A_m$, in which the (configuration) variables $x_1, \cdots, x_m$ are distinct. It denotes a (simple) parallel composition $\|_{x_i} A_i$. In describing the semantics we shall sometimes write Γ for the parallel composition it denotes.

4.2 Configuration expressions

Configuration expressions denote finite configurations of games in an environment. A typing judgement $\Gamma \vdash p : B$ for a configuration expression p in a type environment Γ denotes an affine map from Γ to B. In particular, the judgement $\Gamma, x : A \vdash x : A$ denotes the partial map of event structures projecting to the single component A. The special case $x : A \vdash x : A$ denotes the identity map.

We shall allow configuration expressions to be built from affine maps $f = (f_0, f_1) : A \to_a B$ in the judgement $\Gamma, x : A \vdash fx : B$ and the equivalent judgement $\Gamma, x : A \vdash f_0 \cup f_1 x : B$. In particular, f_1 may be completely undefined, allowing *configuration expressions* to be built from constant configurations, as *e.g.* in the judgement for the empty configuration $\Gamma \vdash \varnothing : A$ or a singleton configuration $\Gamma \vdash \{a\} : A$ when a is an initial event of A. The expression $\{a\} \cup x'$ associated with the judgement

$$\Gamma, x' : A/a \vdash \{a\} \cup x' : A\,,$$

where a is an initial event of A, is used later in the transition semantics.

The three inductive rules for configuration expressions are as follows, where $\Gamma^\perp$ is $x_1 : A_1^\perp, \cdots, x_m : A_m^\perp$:

$$\frac{\Gamma \vdash p : A_j}{\Gamma \vdash jp : \Sigma_{i \in I} A_i} \quad j \in I \qquad \frac{\Gamma \vdash p : A \qquad \Delta \vdash q : B}{\Gamma, \Delta \vdash (p, q) : A \| B} \qquad \frac{\Gamma \vdash p : B}{\Gamma^\perp \vdash p : B^\perp}$$

For a sum $\Sigma_{i \in I} A_i$, the left-hand rule gives configuration expressions jp where $j \in I$ and p is a configuration expression of type A_j: In the central rule for simple parallel composition we exploit the fact that configurations of simple parallel compositions are essentially pairsof configurations of the components. Finally, in the right-hand rule configurations of $B^\perp$ can be taken to be the same as configurations of B.

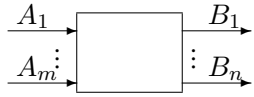
4.3 Terms for strategies

Terms denoting strategies have typing judgements:

$$x_1 : A_1, \cdots, x_m : A_m \vdash \ t \ \dashv y_1 : B_1, \cdots, y_n : B_n \ ,$$

where all the variables are distinct, interpreted as a strategy from the game $x_1 : A_1, \cdots, x_m : A_m$ denotes to the game $y_1 : B_1, \cdots, y_n : B_n$ denotes.

We can think of the term t as a box with input and output wires for the typed variables:



Duality The duality of input and output is caught by the rules:

$$\frac{\Gamma, x : A \vdash t \dashv \Delta}{\Gamma \vdash t \dashv x : A^\perp, \Delta} \qquad \frac{\Gamma \vdash t \dashv x : A, \Delta}{\Gamma, x : A^\perp \vdash t \dashv \Delta}$$

Composition The composition of strategies is described in the rule

$$\frac{\Gamma \vdash t \dashv \Delta \qquad \Delta \vdash u \dashv \mathrm{H}}{\Gamma \vdash \exists \Delta. \, [\, t \parallel u \,] \dashv \mathrm{H}}$$

which, in the picture of strategies as boxes, joins the output wires of one strategy to input wires of the other. Note that the simple parallel composition of strategies arises as a special case when Δ is empty.

Nondeterministic sum We can form the nondeterministic sum of strategies of the same type:

$$\frac{\Gamma \vdash t_i \dashv \Delta \quad i \in I}{\Gamma \vdash \big[\big]_{i \in I} t_i \dashv \Delta}$$

We shall use $\perp$ for the empty nondeterministic sum, when the rule above specialises to $\Gamma \vdash \perp \dashv \Delta$. The term $\perp$ denotes the minimum strategy in the game $\Gamma^\perp \| \Delta$.

Pullback We can form the pullback of two strategies of the same type:

$$\frac{\Gamma \vdash t_1 \dashv \Delta \quad \Gamma \vdash t_2 \dashv \Delta}{\Gamma \vdash t_1 \wedge t_2 \dashv \Delta}$$

In the case where t_1 and t_2 denote the respective strategies $\sigma_1 : S_1 \to \Gamma^\perp \| \Delta$ and $\sigma_1 : S_1 \to \Gamma^\perp \| \Delta$, the strategy $t_1 \wedge t_2$ denotes the pullback of σ_1 and σ_2. Informally, such a strategy acts as the two component strategies agree to act.

Hom-set The hom-set rule is a powerful way to lift affine maps or relations expressed in terms of cospans of affine maps to strategies. Write $p[\varnothing]$ for the substitution of the empty configuration $\varnothing$ for all configuration variables appearing in a configuration expression p. The hom-set rule

$$\frac{\Gamma \vdash p' : C \qquad \Delta \vdash p : C}{\Gamma \vdash p \sqsubseteq_C p' \dashv \Delta} \quad p[\varnothing] \sqsubseteq_C p'[\varnothing]$$

introduces a term standing for the hom-set $(\mathcal{C}(C), \sqsubseteq_C)(p, p')$. It relies on configuration expressions p, p' and their typings. If $\Delta \vdash p : C$ denotes the affine map $g = (g_0, g_1)$ and $\Gamma \vdash p' : C$ denotes the affine map $f = (f_0, f_1)$, the side condition of the rule ensures that $g_0 \sqsubseteq_C f_0$. A term for copy-cat arises as a special case of the hom-set rule: $x : A \vdash y \sqsubseteq_A x \dashv y : A$.

The hom-set rule is very expressive — see Section 4.4. The precise definition of the strategy which the hom-set rule yields is given in the next section.

Duplication Duplication terms are described by the rule

$$\frac{\Gamma \vdash p : C \quad \Delta_1 \vdash q_1 : C \quad \Delta_2 \vdash q_2 : C}{\Gamma \vdash \delta_C(p, q_1, q_2) \dashv \Delta_1, \Delta_2}$$

provided $p[\varnothing], q_1[\varnothing], q_2[\varnothing]$ is balanced in the sense of Section 3.2. The term for the duplication strategy is, in particular, $x : A \vdash \delta_A(x, y_1, y_2) \dashv y_1 : A, y_2 : A$.

4.3.1 Hom-set terms: semantics

The definition of the strategy which $\Gamma \vdash p \sqsubseteq_C p' \dashv \Delta$ denotes is quite involved. We first simplify notation. W.l.o.g. assume $\Delta \vdash p : C$ and $\Gamma \vdash p' : C$ — using duality we can always rearrange the environment to achieve this. Write A for the denotation of the environment Γ and B for the denotation of Δ. Let $\Delta \vdash p : C$ and $\Gamma \vdash p' : C$ denote respectively the affine maps $g = (g_0, g_1) : B \to_a C$ and $f = (f_0, f_1) : A \to_a C$. Note that we have that $g_0 \sqsubseteq_C f_0$ from the typing of $p \sqsubseteq_C p'$. We build the strategy out of a rigid family $\mathcal{Q}$ with elements as follows. First, define a pre-element to be a finite preorder comprising a set $\{1\} \times x \cup \{2\} \times y$, for which $x \in \mathcal{C}(A^\perp)$ & $y \in \mathcal{C}(B)$ & $gy \sqsubseteq_c fx$, with order that induced by $\leq_{A^\perp}$ on x, $\leq_B$ on y, with additional causal dependencies

$$(1, a) \leq (2, b) \text{ if } f_1(a) = g_1(b) \quad \& \ b \text{ is } +\text{ve, and}$$
$$(2, b) \leq (1, a) \text{ if } f_1(a) = g_1(b) \quad \& \ b \text{ is } -\text{ve.}$$

As elements of the rigid family $\mathcal{Q}$ we take those pre-elements for which the order $\leq$ is a partial order (*i.e.* is antisymmetric). The elements of $\mathcal{Q}$ are closed under rigid inclusions, so $\mathcal{Q}$ forms a rigid family. We now take $S =_{\text{def}} \Pr(\mathcal{Q})$; the events of S (those elements of $\mathcal{Q}$ with a top event) map to their top events in $A^\perp \| B$ from where they inherit polarities. This map can be checked to be a strategy: innocence follows directly from the construction, while receptivity follows from the constraint that $gy \sqsubseteq_c fx$.

It is quite easy to choose an example where antisymmetry fails in a pre-element, in other words, in which the preorder is not a partial order. However, when either p or p' is just a variable, no nontrivial causal loops are introduced and all pre-elements are elements. More generally, if one of p or p' is associated with a partial rigid map (*i.e.* a map which preserves causal dependency when defined), then no nontrivial causal loops are introduced and all pre-elements are elements.

4.3.2 Duplication terms: semantics

Consider now the semantics of a term $\Gamma \vdash \delta_C(p, q_1, q_2) \dashv \Delta$. W.l.o.g. we may assume that the environment is arranged so $\Delta \equiv \Delta_1, \Delta_2$ with judgements $\Gamma \vdash p : C$, $\Delta_1 \vdash q_1 : C$ and $\Delta_2 \vdash q_2 : C$. To simplify notation assume the latter judgements for configuration expressions denote the respective affine maps $f = (f^0, f^1) : A \to_a C$, $g_1 = (g_1^0, g_1{}^1) : B_1 \to C$ and $g_2 = (g_2^0, g_2{}^1) : B_2 \to C$. From the typing of $\delta_C(p, q_1, q_2)$ we have that (f^0, g_1^0, g_2^0) forms a balanced triple in C. We build the strategy out of a rigid family $\mathcal{Q}$ with elements as follows. We construct pre-elements from $x \in \mathcal{C}(A^\perp)$, $y_1 \in \mathcal{C}(B_1)$ and $y_2 \in \mathcal{C}(B_2)$ where $(fx, g_1 y_1, g_2 y_2)$ is a balanced triple in C with a choice function χ. There are three kinds of elements of x:

$$
\begin{aligned}
x^- &= \{a \in x \mid pol_{A^\perp}(a) = -\}, \\
x_0^+ &= \{a \in x \mid pol_{A^\perp}(a) = + \ \&\ f^1(a) \in g^0_{\chi(f^1(a))}\} \text{ and} \\
x_1^+ &= \{a \in x \mid pol_{A^\perp}(a) = + \ \&\ f^1(a) \in g^1_{\chi(f^1(a))} y_{\chi(f^1(a))}\}
\end{aligned}
$$

We define a typical pre-element to be a finite preorder on the set

$$\{0\} \times (x^- \cup x_1^+ \cup \{(\chi(f^1(a)), a) \mid a \in x_0^+\}) \cup \{1\} \times y_1 \cup \{2\} \times y_2,$$

with order that induced by that of the game $A^\perp \| B_1 \| B_2$ with additional causal dependencies

$$
\begin{aligned}
(0, a) &\leq (1, b) \text{ if } f^1(a) = g_1^1(b) \ \&\ \text{ b is +ve in } B_1, \\
(0, a) &\leq (2, b) \text{ if } f^1(a) = g_2^1(b) \ \&\ \text{ b is +ve in } B_2, \text{ and} \\
(\chi(f^1(a)), b) &\leq (0, a) \text{ if } a \in x_1^+ \ \&\ f^1(a) = g^1_{\chi(f^1(a))}(b),
\end{aligned}
$$

for b $-$ve in $B_{\chi(f^1(a))}$. As elements of the rigid family $\mathcal{Q}$ we take those pre-elements for which the order $\leq$ is a partial order (*i.e.* is antisymmetric). Once $\mathcal{Q}$ is checked to be a rigid family, we can take $S =_{\text{def}} \Pr(\mathcal{Q})$; the events of S map to the events in the game $A^\perp \| B_1 \| B_2$ associated with their top events, from where they inherit polarities. This map defines the strategy denoting the original duplication term.

4.4 Expressivity

The terms for strategies are surprisingly expressive and potentally rich in laws, able to support a form of equational reasoning, that we can only touch on here. For example the Frobenius algebra associated with duplication immediately yields laws. Other laws capture basic facts about the Scott order. For instance, assuming $z \sqsubseteq x, y$ in $\mathcal{C}(A)$, we have $y \sqsubseteq_A x$ iff $y/z \sqsubseteq_{A/z} x/z$.

As we shall see, we can derive the laws expected of a recursion operator provided the recursion involves a homorphism w.r.t. the duplication comonad, and this fact too we could hope to derive. Some of the reasoning can be made diagrammatic, using the techniques of string diagrams.

Hom-set terms provide many basic strategies. The denotation of $x : A \vdash \varnothing \sqsubseteq_A \varnothing \dashv y : B$ is the strategy in the game $A^\perp \| B$ given by the identity map $\mathrm{id}_{A^\perp \| B} : A^\perp \| B \to A^\perp \| B$. The denotation of $\vdash y \sqsubseteq_A \varnothing \dashv y : A$ is $\bot_A$, the minimum strategy in the game A comprising just the initial −ve events of A.

The judgement $x : A_j \vdash y \sqsubseteq_{\Sigma_{i\in I} A_i} jx \dashv y : \Sigma_{i\in I} A_i$ denotes the injection strategy: its application to a strategy in A_j fills out the strategy according to the demands of receptivity to a strategy in $\Sigma_{i\in I} A_i$. Its converse $x : \Sigma_{i\in I} A_i \vdash jy \sqsubseteq_{\Sigma_{i\in I} A_i} x \dashv y : A_j$ applied to a strategy of $\Sigma_{i\in I} A_i$ projects the strategy to a strategy in A_j.

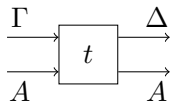
More is obtained by combining hom-set with other operations such as composition. Assume $\vdash t \dashv y : B$. When $f : A \to B$ is a map of event structures with polarity, the composition $\vdash \exists y : B. [\, t \parallel fx \sqsubseteq_B y \,] \dashv x : A$ denotes the pullback $f^*\sigma$ of the strategy σ denoted by t across the map $f : A \to B$. In the case where a map of event structures with polarity $f : A \to B$ is innocent, the composition $\vdash \exists x : A. [\, y \sqsubseteq_B fx \parallel t \,] \dashv y : B$ denotes the 'relabelling' $f_!\sigma$ of the strategy σ denoted by t.

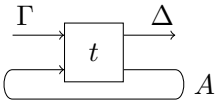
A great deal is achieved through basic manipulation of the input and output "wiring" afforded by the hom-set rules and input-output duality. For instance, to achieve the effect of *lambda abstraction*: via the hom-set rule we obtain

$$x : A^\perp, y : B \vdash z \sqsubseteq_{A\|B} (x, y) \dashv z : A^\perp \| B \ ,$$

which joins two inputs to a common output, whence:

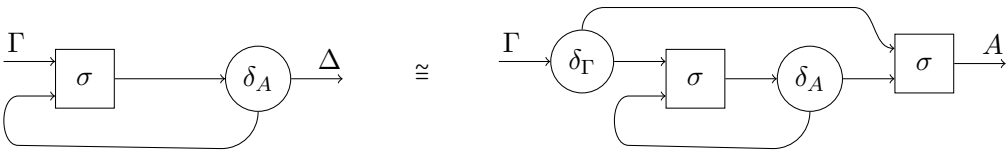
$$\frac{\dfrac{\Gamma, x : A \vdash t \dashv y : B}{\Gamma \vdash t \dashv x : A^\perp, y : B} \quad \dfrac{\vdots}{x : A^\perp, y : B \vdash z \sqsubseteq_{A^\perp \| B} (x, y) \dashv z : A^\perp \| B}}{\Gamma \vdash \exists x : A^\perp, y : B. [\, t \parallel z \sqsubseteq_{A^\perp \| B} (x, y)) \,] \dashv z : A^\perp \| B}$$

A *trace*, or feedback, operation is another consequence of such "wiring." Given a strategy $\Gamma, x : A \vdash t \dashv y : A, \Delta$ represented by the diagram  we obtain $\Gamma, \Delta^\perp \vdash t \dashv x : A^\perp, y : A$, which, post-composed with the term $x : A^\perp, y : A \vdash x \sqsubseteq_A y \dashv$ denoting the copy-cat strategy $\gamma_{A^\perp}$, yields $\Gamma \vdash \exists x : A^\perp, y : A. [\, t \parallel x \sqsubseteq_A y \,] \dashv \Delta$, representing its trace:



The composition introduces causal links from the +ve events of $y : A$ to the −ve events of $x : A$, and from the +ve events of $x : A$ to the −ve events of $y : A$ — these are the usual links of copy-cat $\gamma_{A^\perp}$ as seen from the left of the turnstyle. This trace coincides with the feedback operation which has been used in the semantics of nondeterministic dataflow (where only games comprising solely Player moves are needed) [13].

Recursive definitions can be achieved from trace with the help of duplication. For those strategies which respect δ, *i.e.*$\delta_A \odot \sigma \cong (\sigma \| \sigma) \odot \delta_{\Gamma \| A}$, and in particular for strategies which are homomorphisms between δ-comonoids, the recursive definition does unfold in the way expected, in the sense that the recursive definition is isomorphic to its unfolding:



This follows as a general fact from the properties of trace monoidal categories and the string-diagram reasoning they support. However, not all strategies are homomorphisms between δ-comonoids, characterised in the following theorem.

Theorem 4.1 *A strategy $\sigma : S \to A^\perp \parr B$ respects δ iff*

- *components $\sigma_1 : S \to A^\perp$ and $\sigma_2 : S \to B$ preserve causal dependency when defined,*
- *σ_1 reflects configurations of $A^\perp$, i.e. if $x \subseteq S$ is such that $\sigma_1 x \in \mathcal{C}(A^\perp)$ then $x \in \mathcal{C}(S)$, and*
- *for every +ve event $s \in S$ such that $\sigma(s) \in A^\perp$ we have the number of −ve events of $\sigma_2[s]$ equals the number of +ve events of $\sigma_1[s]$.*

In this case, σ also respects counits, i.e. $\bot_B \odot \sigma \cong \bot_A$.

5 A process perspective

The operations on strategies have much in common with the operations of process algebra and can be seen as forming the basis of a higher-order process language. However, from the perspective of concurrent processes, we must address several issues: its *operational semantics*, a suitable form of *equivalence* and *expressivity*. These require we examine the effects of synchronization and the internal, neutral events it produces, more carefully. Composition of strategies can introduce deadlock which is presently hidden. This, for example, affects the reliability of winning strategies; presently a strategy may be deemed winning and yet possibly deadlock before reaching a winning configuration. The next example illustrates how hidden deadlocks may be created in a composition of strategies.

Example 5.1 (i) Deadlock may arise in a composition $\tau \odot \sigma$ through $\sigma : A \mapsto\!\!\!\!\rightarrow B$ and $\tau : B \mapsto\!\!\!\!\rightarrow C$ imposing incompatible causal dependencies between events in B. For instance B may contain two concurrent events of opposite polarities $b_1 = \ominus$ and $b_2 = \oplus$. The strategy σ may impose the causal dependency $s_1 \rightarrowtriangle s_2$ between occurrences of b_1 and b_2 respectively. From the point of view of strategy τ, the game B has changed polarity to $B^\perp$ and τ may impose the reverse causal dependency $s_2 \rightarrowtriangle s_1$ between occurrences of b_2 and b_1 respectively.
(ii) Composition of strategies may hide computation which is stuck. For games $B = \oplus \| \oplus$ and $C = \oplus$, assume strategy $\sigma_1 : \varnothing \mapsto\!\!\!\!\rightarrow B$ nondeterministically chooses the right or left move in B, strategy $\sigma_2 : \varnothing \mapsto\!\!\!\!\rightarrow B$ chooses just the right move in B, while strategy $\tau : B \mapsto\!\!\!\!\rightarrow C$ yields output in C if it gets the right event of B as input. The two strategy compositions $\tau \odot \sigma_1$ and $\tau \odot \sigma_2$ are indistinguishable. □

5.1 Partial strategies

To treat such phenomena explicitly and in order to obtain a transition semantics, we extend strategies with neutral events. Extend event structures with polarity to include a neutral polarity 0; as before, maps preserve polarities when defined. However within games we shall still assume that all events have +ve or −ve polarity.

Definition 5.2 A *partial strategy from* a game A *to* a game B comprises a total map $\sigma : S \to A^\perp \| N \| B$ of event structures with polarity (in which S may also have neutral events) where
(i) N is an event structure consisting solely of neutral events;
(ii) σ is *receptive*, *i.e.* if $\sigma x \overset{c}{\multimap}$ in $\mathcal{C}(S)$ with c −ve, then $x \overset{s}{\multimap}$ and $\sigma(s) = c$, for some unique $s \in S$;
(iii) in the partial-total factorization of the composition of σ with the projection $A^\perp \| N \| B \to A^\perp \| B$, drawn below, the defined part σ_0 is a strategy.

$$\begin{array}{ccc} S & \longrightarrow & S_0 \\ {\scriptstyle\sigma}\downarrow & & \downarrow{\scriptstyle\sigma_0} \\ A^\perp \| N \| B & \rightarrowtriangle & A^\perp \| B \end{array}$$

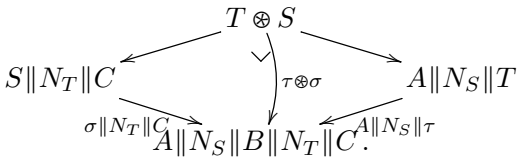
Partial strategies in a game A correspond to partial strategies from the empty game to A. Strategies between games correspond to those partial strategies in which the neutral events N form the empty event structure.

It may seem odd that partial strategies are total maps. Why have we not taken a partial strategy to be undefined on events which are sent to N? Because such partial maps do not behave well under pullback, and this would complicate the definition of composition and spoil later results such as that the pullback of a partial strategy is a partial strategy. With our choice of definition we are able to localise neutral events to the games over which they occur; with the alternative definition, different forms of undefined would become conflated.

5.2 Operations on partial strategies

The operations on strategies extend and give an interpretation of the language of Section 4 in terms of partial strategies. The defined parts of the operations on partial strategies coincide with the operations on the defined parts.

We can compose two partial strategies $\sigma : S \to A^\perp \| N_S \| B$ and $\tau : T \to B^\perp \| N_T \| C$ by pullback. Ignoring polarities temporarily, and padding with identity maps, we obtain $\tau \circledast \sigma$ via the pullback

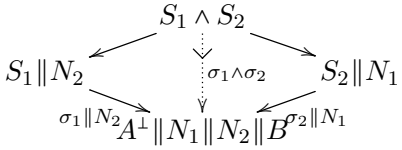


once we reinstate polarities and make the events of B neutral. Receptivity of $\tau \circledast \sigma$ follows directly from that of σ and τ. That the defined part of $\tau \circledast \sigma$ is a strategy follows once it is shown that the defined part of the composite $T \circledast S \xrightarrow{\tau \circledast \sigma} A^\perp \| (N_S \| B \| N_T) \| C \longrightarrow A^\perp \| C$ is isomorphic to $\tau_0 \odot \sigma_0$, the composition of the defined parts of σ and τ.

With partial strategies we no longer generally have that composition with copycat yields the same strategy up to isomorphism: there will generally be extra neutral events introduced through synchronizations. However, as demonstrated in Section 6, a bicategory may be recovered through the use of may/must equivalence.

Let $\sigma_i : S_i \to A^\perp \| N_i \| B$, where $i \in I$, be a family of partial strategies. Their *sum* is the partial strategy $[]_{i \in I} \sigma_i : S \to A^\perp \| (\|_{i \in I} N_i) \| B$. Its events are obtained as the disjoint union of the S_i but where the initial $-$ve events are identified to maintain receptivity; they map under $[]_{i \in I} \sigma_i$ as directed by the component maps σ_i. Causal dependency is inherited from the components S_i with a finite subset of events consistent iff its down-closure contains $+$ve events from at most one S_i. As such, the nondeterministic sum only commits to a component through the occurrence of a positive event: from the perspective of tracking potential deadlocks, it is not necessary to view neutral events as committing to a particular component since, in isolation, a neutral event cannot introduce deadlock when composed with a counterstrategy due to receptivity.

The *pullback* of partial strategies $\sigma_1 : S_1 \to A^\perp \| N_1 \| B$ and $\sigma_2 : S_2 \to A^\perp \| N_2 \| B$ is obtained as follows:



5.3 Transition semantics

We now turn to the transition semantics for partial strategies, which is presented in Figure 1. For brevity, the rules presented require the left-hand environment in the

triple denoting a strategy to be empty; this can always be achieved using the rules for duality. In the transition rules, we write $t \dashv \Delta$ instead of $\varnothing \vdash t \dashv \Delta$. Transitions are associated with two kinds of actions, either an action o associated with a hidden neutral action $t \dashv \Delta \xrightarrow{o} t' \dashv \Delta$ or an initial event located in the environment $t \dashv x : A, \Delta \xrightarrow{x:a:x'} t' \dashv x' : A/a, \Delta$. Notice that a neutral action leaves the types unchanged but may affect the term. An action $x : a : x'$ is associated with an initial event $ev(x : a : x') =_{\text{def}} x : a$ at the x-component of the environment. On its occurrence, the component of the environment $x : A$ is updated to $x' : A/a$ in which x', a fresh resumption variable, stands for the configuration remaining in the remaining game A/a. Say an action $x : a : x'$ is +ve/−ve according as a is +ve/−ve. In the ruless for composition, we use α for o or an action of the form $x : a : x'$ where x is in the domain of Γ and use β for o or an action of the form $y : b : y'$ where y is in the domain of H.

In typed judgements of $\delta_C(p, q_1, q_2)$, a variable can appear free in at most one of the configuration expressions p, q_1 and q_2. Write, for example, $y \in \mathrm{fv}(p)$ for y is free in p, and $q_1(y : b) \in p[\varnothing]$ to mean the image of b under the map q_1 denotes is in the configuration denoted by $p[\varnothing]$.

We now establish the correspondence of the operational semantics with our view of terms as denoting strategies. Given a strategy $\sigma : S \to \Delta$ and an initial event s of S, we write $\sigma/s : S/e \to \Delta/\sigma(s)$ for the strategy obtained by restricting σ to S/s.

Theorem 5.3 *Assume certain primitive strategies $\varnothing \vdash \sigma_0 \dashv \Delta$, so as a map, $\sigma_0 : S \to \Delta$, for which we assume rules,*

$$\frac{}{\sigma_0 \dashv \Delta \xrightarrow{\epsilon} \sigma_0' \dashv \Delta'} s \text{ is initial in } S \ \& \ \sigma_0(s) = ev(\epsilon).$$

Then, derivations in the operational semantics

$$\frac{\vdots}{t \dashv \Delta \xrightarrow{\epsilon} t' \dashv \Delta',}$$

up to α-equivalence, in which t denotes the partial strategy $\sigma : S \to \Delta$, are in 1-1 correspondence with initial events s in S such that $\sigma(s) = ev(\epsilon)$ when $ev(\epsilon) \neq o$ or s is neutral when $ev(\epsilon) = o$. Furthermore, letting t' denote the partial strategy $\sigma' : S' \to \Delta'$, the strategies σ/s and σ' are isomorphic.

6 May and Must equivalence

We now study 'may' and 'must' equivalence and how it may be used to recover bicategorical structure on strategies equipped with *stopping configurations*.

Consider three partial strategies in a game comprising a single +ve event. For this discussion it will be sufficient to consider just the event structures of the partial strategies:

$$S_1 \quad \oplus \qquad S_2 \quad \circledcirc \rightarrowtriangle \oplus \qquad S_3 \quad \circledcirc \sim \circledcirc \rightarrowtriangle \oplus$$

Composition:

$$\frac{t \dashv y : B, \Delta, \Gamma \xrightarrow{y:b:y'} t' \dashv y' : B/b, \Delta, \Gamma \qquad u \dashv y : B^\perp, \Delta^\perp, \mathrm{H} \xrightarrow{y:b:y'} u' \dashv y' : B^\perp/b, \Delta^\perp, \mathrm{H}}{\exists y : B, \Delta. [\, t \parallel u \,] \dashv \Gamma, \mathrm{H} \xrightarrow{o} \exists y' : B/b, \Delta. [\, t' \parallel u' \,] \dashv \Gamma, \mathrm{H}}$$

$$\frac{t \dashv \Gamma, \Delta \xrightarrow{\alpha} t' \dashv \Gamma', \Delta}{\exists \Delta. [\, t \parallel u \,] \dashv \Gamma \xrightarrow{\alpha} \exists \Delta. [\, t' \parallel u \,] \dashv \Gamma'} \qquad \frac{u \dashv \mathrm{H}, \Delta^\perp \xrightarrow{\beta} u' \dashv \mathrm{H}', \Delta^\perp}{\exists \Delta. [\, t \parallel u \,] \dashv \mathrm{H} \xrightarrow{\beta} \exists \Delta. [\, t \parallel u' \,] \dashv \mathrm{H}'}$$

Hom-sets: Assuming a is an initial event of A for which $p[\{a\}/x][\varnothing] \sqsubseteq_C p'[\{a\}/x][\varnothing]$,

$$p \sqsubseteq_C p' \dashv x : A, \Delta \xrightarrow{x:a:x'} p[\{a\} \cup x'/x] \sqsubseteq_C p'[\{a\} \cup x'/x] \dashv x' : A/a, \Delta$$

Above, the variable x will only appear in one of p and p'.

Sum of partial strategies:

$$\frac{t_j \dashv \Delta \xrightarrow{\epsilon} t_j' \dashv \Delta'}{\bigsqcap_{i \in I} t_i \dashv \Delta \xrightarrow{\epsilon} t_j' \dashv \Delta'} j \in I \;\&\; \epsilon \text{ is +ve} \qquad \frac{t_i \dashv \Delta \xrightarrow{\epsilon} t_i' \dashv \Delta' \;\; \forall i \in I}{\bigsqcap_{i \in I} t_i \dashv \Delta \xrightarrow{\epsilon} \bigsqcap_{i \in I} t_i' \dashv \Delta} \epsilon \text{ is } -\text{ve}$$

$$\frac{t_j \dashv \Delta \xrightarrow{o} t_j' \dashv \Delta'}{\bigsqcap_{i \in I} t_i \dashv \Delta \xrightarrow{o} \bigsqcap_{i \in I} t_i' \dashv \Delta} j \in I, \text{ where } t_i' = t_i \text{ if } i \neq j$$

Pullback:

$$\frac{t_1 \dashv \Delta \xrightarrow{o} t_1' \dashv \Delta}{t_1 \wedge t_2 \dashv \Delta \xrightarrow{o} t_1' \wedge t_2 \dashv \Delta} \qquad \frac{t_2 \dashv \Delta \xrightarrow{o} t_2' \dashv \Delta}{t_1 \wedge t_2 \dashv \Delta \xrightarrow{o} t_1 \wedge t_2' \dashv \Delta}$$

$$\frac{t_i \dashv \Delta \xrightarrow{z:c:z'} t_i' \dashv \Delta' \;\; \forall i \in \{1,2\}}{t_1 \wedge t_2 \dashv \Delta \xrightarrow{z:c:z'} t_1' \wedge t_2' \dashv \Delta}$$

Duplication:

$$\delta_C(p, q_1, q_2) \dashv x : A, \Delta \xrightarrow{x:a:x'} \delta_C(p, q_1, q_2)[\{a\} \cup x'/x] \dashv x' : A/a, \Delta$$

if either a is an initial $-$ve event of A, or a is an initial +ve event of A and there exists $i \in \{1,2\}$ s.t. either

- $x \in \mathrm{fv}(q_i)$ and $q_i(x : a) \in p[\varnothing]$ or
- $x \in \mathrm{fv}(p)$ and $p(x : a) \in q_i[\varnothing]$.

Fig. 1. Transition semantics

Neutral events are drawn as ⊛. Conflict between pairs of events (meaning that there is no set in the consistency relation containing them both) is drawn as $\sim$. All three partial strategies have the same strategy as their defined parts. However, from the point of view of observing the move in the game, the first two partial strategies differ from the third. In a maximal play both S_1 and S_2 will result in the observation of the single move of the game. However, in S_3 one maximal play is that in which the leftmost neutral event has occurred, in conflict with observing the single move of the game.

We follow [5] in making these ideas precise. For configurations x, y of an event structure with polarity which may have neutral events write $x \subseteq^p y$ to mean $x \subseteq y$ and all events of $y \setminus x$ have polarity + or 0. We write $\subseteq^0$ to mean the inclusion involves only neutral events

Definition 6.1 Let σ be a partial strategy in a game A. Let $\tau : T \to A^\perp \| \oplus$ be a 'test' strategy from A to a the game consisting of a single Player move ✓.

Say σ *may pass* τ iff there exists $x \in \mathcal{C}^\infty(T \circledast S)$ with image $(\tau \circledast \sigma)x$ containing ✓.

Say σ *must pass* τ iff all $x \in \mathcal{C}^\infty(T \circledast S)$ which are $\subseteq^p$-maximal have image $(\tau \circledast \sigma)x$ containing ✓.

Say two partial strategies are *'may' ('must') equivalent* iff the tests they may (respectively, must) pass are the same.

Two partial strategies with the same strategy as their defined part are 'may' equivalent but need not be 'must' equivalent. 'Must' inequivalence is lost in moving from partial strategies to strategies. Moreover, as we have seen, partial strategies lack identities w.r.t. composition, so they do not even form a bicategory. Fortunately, for 'may' and 'must' equivalence it is not necessary to use partial strategies; it is sufficient to carry with a strategy the extra structure of 'stopping' configurations (= images of p-maximal configurations in a partial strategy). Composition and copy-cat on strategies extend to composition and copy-cat on strategies with stopping configurations, while maintaining a bicategory, in the following way.

First, to deal with races, we are forced to introduce a refinement of the Scott order. We write $x \lhd^* y$ for the transitive closure of the relation $\lhd \subseteq \mathcal{C}^\infty(S) \times \mathcal{C}^\infty(S)$ defined as

$$x \lhd y \iff x \sqsubseteq y \text{ and } y \parallel x \text{ is } +\text{-maximal in } \mathbb{CC}_S.$$

On race-free games, $\lhd$ is the identity relation on $\mathcal{C}^\infty(S)$.

Let $\sigma : S \to A^\perp \| N \| B$ be a partial strategy from a game A to a game B. Recall its associated partial-total factorization has defined part a strategy $\sigma_0 : S_0 \to A^\perp \| B$. Define the (possibly) *stopping* configurations in $\mathcal{C}^\infty(S_0)$ to be

$$\mathrm{Stop}(\sigma) =_{\mathrm{def}} \downarrow \{ d\,x \mid x \in \mathcal{C}^\infty(S) \text{ is } p\text{-maximal} \},$$

where $d : S \to S_0$ is the partial map that is undefined where σ is undefined and $\downarrow S$ is the down-closure of S for the order $\lhd^*$. Note that $\mathrm{Stop}(\sigma)$ will include all the +-maximal configurations of S_0: any +-maximal configuration y of S_0 is the image under p of its down-closure $[y]$ in S, and by Zorn's lemma this extends (necessarily

by neutral events) to a maximal configuration x of S with image y under d; by maximality, if $x \xrightarrow{s} \subset$ then s cannot be neutral, nor can it be +ve as this would violate the +-maximality of y.

The operation $\mathrm{St} : \sigma \mapsto (\sigma_0, \mathrm{Stop}(\sigma))$ above, from partial strategies to strategies with stopping configurations, motivates the following definitions.

A *strategy with stopping configurations* in a game A comprises a strategy $S \to A$ together with a subset $M_S \subseteq \mathcal{C}^\infty(S)$ which is the down-closure with respect to $\lhd^*$ of a set of +-maximal configurations. As usual, a *strategy with stopping configurations* from a game A to game B is a strategy with stopping configurations in the game $A^\perp \| B$. We can define 'may' and 'must' testing of strategies with stopping configurations analogously to above.

Given two strategies with stopping configurations $\sigma : S \to A^\perp \| B$, M_S and $\tau : T \to B^\perp \| C$, M_T we define their composition by $(\tau, M_T) \odot (\sigma, M_S) =_{\mathrm{def}} (\tau \odot \sigma, M_T \odot M_S)$ where $x \in M_T \odot M_S$ iff

$$\exists z \in \mathcal{C}^\infty(T \circledast S).\ [x]_{T \circledast S} \subseteq^0 z \ \&\ \Pi_1 z \in M_S \ \&\ \Pi_2 z \in M_T\,.$$

The stopping configurations of copy-cat are obtained as for any other strategy, and in particular $M_{CC_A} = \{y \parallel x \mid x, y \in \mathcal{C}^\infty(A) \ \&\ x \lhd^* y\}$.

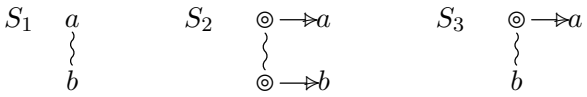
Proposition 6.2 *$\gamma_A, M_{\mathbb{C}\mathbb{C}_A}$ is an identity w.r.t. the composition on strategies with stopping configurations.*

Proposition 6.3 *Let σ be a partial strategy from A to B and τ a partial strategy from B to C. Then*

$$\mathrm{Stop}(\tau \circledast \sigma) = \mathrm{Stop}(\tau) \odot \mathrm{Stop}(\sigma)\,.$$

Corollary 6.4 *A partial strategy σ 'may' (respectively 'must') pass a test τ iff $St(\sigma)$ 'may' ('must') pass τ. The operation St preserves 'may' and 'must' equivalence.*

Example 6.5 It is tempting to think of neutral events as behaving like the internal "tau" events of CCS [10]. However, in the context of strategies they behave rather differently. Consider three partial strategies, over a game comprising of just two concurrent +ve events, say a and b. The partial strategies have the following event structures in which we have named events by the moves they correspond to in the game:



All three become isomorphic under St so are 'may' and 'must' equivalent to each other. □

Strategies with stopping configurations inherit the structure of a bicategory from strategies. We can interpret the metalanguage directly in terms of strategies with stopping configurations in such a way that the denotation of a term as a strategy with stopping configurations is the image under St of its denotation as a partial strategy. To achieve this, we specify the stopping configurations of both the sum

and pullback of strategies.

For the sum of strategies $[]_{i\in I}\,\sigma_i$ with stopping configurations σ_i, a configuration of the sum is stopping iff it is the image of a stopping configuration under the injection from a component.

Consider strategies $\sigma : S \to A$ and $\tau : T \to A$ with stopping configurations M_S and M_T respectively. Let their pullback be denoted by $\sigma \wedge \tau : P \to A$ with projection morphisms $\pi_1 : P \to S$ and $\pi_2 : P \to T$. A configuration of P is defined to be stopping iff there exist x_1, x_2 such that $\pi_1 x \subseteq^+ x_1$ and $\pi_2 x \subseteq^+ x_2$ and $x_1 \in M_S$ and $x_2 \in M_T$, and furthermore there exists a partition $x^+ = Y_1 \cup Y_2$ satisfying $x_i \cap Y_i = \varnothing$. The set of stopping configurations of P coincides with the stopping configurations obtained via St from the pullback of partial strategies.

The treatment of winning strategies of [3] generalises straightforwardly, with the role of +-maximal configurations replaced by that of stopping configurations.

7 Extensions & concluding remarks

We have seen a range of constructions on concurrent strategies that support a rich higher-order language for them, with a corresponding operational semantics. For the latter a central part has been the introduction of partial strategies. Though composition of a partial strategy with copy-cat does not in general yield the same strategy, we have seen how a bicategory can be obtained that respects the may/must behaviour of partial strategies by using *stopping configurations*.

The bicategorical structure of strategies is largely undisturbed by extensions to *probabilistic* and *quantum* games [18], *imperfect information* [17] and *symmetry* [1] — though compact-closure becomes $*$-autonomy under extensions by *winning conditions* [3] and *pay-off* [4]. The language of strategies is applicable in these extensions, with minor modifications. The constraints of linearity can be alleviated in games with symmetry which support (co)monads for copying [1]. The constructions of the language extend fairly directly to games with symmetry, though to exploit symmetry fully the language needs to be extended to accommodate (co)monads up to symmetry.

References

[1] Castellan, S., P. Clairambault and G. Winskel, *Symmetry in concurrent games*, in: *LICS 2014* (2014).

[2] Cattani, G. L. and G. Winskel, *Profunctors, open maps and bisimulation*, Mathematical Structures in Computer Science **15** (2005), pp. 553–614.

[3] Clairambault, P., J. Gutierrez and G. Winskel, *The winning ways of concurrent games*, in: *LICS '12* (2012).

[4] Clairambault, P. and G. Winskel, *On concurrent games with payoff*, in: *MFPS '13*, number 298 in ENTCS (2013).

[5] De Nicola, R. and M. Hennessy, *Testing equivalences for processes*, Theoretical Computer Science **34** (1984).

[6] Faggian, C. and M. Piccolo, *Partial orders, event structures and linear strategies*, in: *TLCA '09*, LNCS **5608**, 2009.

[7] Hirschowitz, T., *Full abstraction for fair testing in ccs*, in: *CALCO '13*, LNCS **8089**, 2013 .

[8] Honda, K., *Processes and games*, ENTCS **71** (2004), {WRLA} 2002, Rewriting Logic and Its Applications.
URL http://www.sciencedirect.com/science/article/pii/S1571066105825289

[9] Hyland, M., *Some reasons for generalising domain theory*, Mathematical Structures in Computer Science **20** (2010), pp. 239–265.

[10] Milner, R., "Communication and concurrency," Prentice Hall, 1989.

[11] Nygaard, M. and G. Winskel, *Linearity in process languages*, in: *LICS '02* (2002).

[12] Rideau, S. and G. Winskel, *Concurrent strategies*, in: *LICS 2011* (2011).

[13] Saunders-Evans, L. and G. Winskel, *Event structure spans for nondeterministic dataflow*, ENTCS **175** (2007).

[14] Winskel, G., *Event structure semantics for CCS and related languages*, in: *ICALP'82*, LNCS **140** (1982).

[15] Winskel, G., *Relations in concurrency*, in: *LICS '05* (2005).

[16] Winskel, G., *Event structures with symmetry*, Electr. Notes Theor. Comput. Sci. 172: 611-652 (2007).

[17] Winskel, G., *Winning, losing and drawing in concurrent games with perfect or imperfect information*, in: *Festschrift for Dexter Kozen*, LNCS **7230** (2012).

[18] Winskel, G., *Distributed probabilistic and quantum strategies*, in: *MFPS '13*, number 298 in ENTCS (2013).

[19] Winskel, G., *Strategies as profunctors*, in: *FOSSACS '13*, LNCS (2013).